Direct Policy Iteration with Demonstrations

Jessica Chemali, Alessandro Lazaric

Direct Policy Iteration with Demonstrations

Jessica Chemali
Machine Learning Department
Carnegie Mellon University

Alessandro Lazaric
SequeL team
INRIA Lille

Abstract

We consider the problem of learning the optimal policy of an unknown Markov decision process (MDP) when expert demonstrations are available along with interaction samples. We build on classification-based policy iteration to perform a seamless integration of interaction and expert data, thus obtaining an algorithm which can benefit from both sources of information at the same time. Furthermore, we provide a full theoretical analysis of the performance across iterations providing insights on how the algorithm works. Finally, we report an empirical evaluation of the algorithm and a comparison with the state-of-the-art algorithms.

1 Introduction

We study the problem of learning the optimal policy of an unknown Markov decision process (MDP) by reinforcement leaning (RL) when *expert* samples (limited in number and/or quality) are available. In particular, we identify in *approximate policy iteration* (API) [Bertsekas and Tsitsiklis, 1996] the most suitable learning scheme to integrate expert demonstrations. API is an iterative learning scheme where one has access to batch data (i.e., *RL samples*) obtained by a sampling policy. Starting at an arbitrary policy, API goes through two stages at every iteration: 1) *policy evaluation* first computes the value function of the current policy, 2) *policy improvement* computes the greedy policy w.r.t. the value function. At any of the two steps, API may introduce errors that depend on the use of a finite number of samples and the choice of a specific approximation space (e.g., linear regression). Because of such errors, API is not guaranteed to improve policies monotonically at each iteration and, in general, it only converges to a set of policies with bounded performance loss w.r.t. the optimal policy [Munos, 2003]. In this work, we propose to improve such learning scheme by modifying the policy improvement step to bias it towards policies that agree, to some degree, with demonstrated expert actions.

Related work. While most of past literature focused on learning from either RL or expert samples [Ng and Russell, 2000; Abbeel and Ng, 2004; Bagnell and Ross, 2010], there is a recent surge of interest in combining expert demonstrations in RL algorithms [Kim *et al.*, 2013; Piot *et al.*, 2014] or cost functions in imitation learning [Ross and Bagnell, 2014].

Kim *et al.* [2013] proposed API *with demonstrations* (APID) which modifies regularized-LSPI [Farahmand *et al.*, 2009] by adding linear soft max-margin constraints in the policy evaluation step; the constraints require value functions for which expert actions are *greedy*. Kim *et al.* [2013] upper bound the Bellman error of APID at any iteration and empirically show that it outperforms both LSPI and imitation learning algorithms. Although the empirical results are a convincing proof of concept, the theoretical analysis is limited to single-iteration errors and does not provide a clear understanding on how effectively RL and expert data are integrated. In fact, while the objective of LSPI is to compute the value function of the current policy, the constraints favor functions with greedy actions equivalent to the expert's. These two requirements may be contrasting (e.g., the greedy policy of the exact value function may not be similar to the expert) and may point towards solutions which are neither good to estimate the target value nor in satisfying the desired constraints.

Piot *et al.* [2014] build on a similar idea but integrate expert constraints directly into the minimization of the optimal Bellman residual. This integration is very appealing since it "regularizes" the global objective of approximating the optimal value function rather than constraining each iteration of API, in which the constraints and the objective over iterations may not always be coherent. Unfortunately, the resulting optimization problem is non-convex, non-differentiable and biased, and so they have to resort to a non-parametric method where the conditional transition matrix is embedded in RKHS and the minimization is carried out by a boosting algorithm. Although the resulting algorithm, called RLED, outperforms APID in practice, the complexity of the optimization prevents from deriving theoretical guarantees on its performance.

Finally, the integration of expert demonstrations has been investigated in Bayesian model-based RL in [Doshi-Velez *et al.*, 2010], where a non-parametric Bayesian approach is employed to combine model knowledge from the expert trajectories and RL samples obtained from independent exploration.

In this paper, we build on classification-based policy iteration [Lagoudakis and Parr, 2003], notably the *direct policy iteration* (DPI) [Lazaric *et al.*, 2010] algorithm, where policy improvement is replaced by a classification problem. Since imitating the expert policy is intrinsically a classifica-

Algorithm 1 Direct Policy Iteration with Demonstrations

Input: policy space Π, state distribution ρ, expert weight α
Initialize: arbitrary policy $\widehat{\pi}_0 \in \Pi$
for k = 1 to K **do**
  Construct the rollout set $\mathcal{D}^k_{\text{roll}} = \{x_i\}_{i=1}^{N_{RL}},\ x_i \overset{iid}{\sim} \rho$
  for $x_i \in \mathcal{D}^k_{\text{roll}}$ and actions $a \in \mathcal{A}$ **do**
    for j = 1 to M **do**
      Perform a rollout according to policy $\widehat{\pi}_k$
      $R^{\pi_k}_j(x_i, a) = r(x_i, a) + \sum_{t=1}^{H} \gamma^t r(x^t, \pi_k(x^t))$,
    end for
    $\widehat{Q}^{\pi_k}(x_i, a) = \frac{1}{M} \sum_{j=1}^{M} R^{\pi_k}_j(x_i, a)$
    Build RL training set $\mathcal{D}^k_{RL} = \mathcal{D}^k_{RL} \cup \{x_i, a^*_i, w_i\}$
  end for
  Build expert training set $\mathcal{D}^k_E = \{z_j, a_j, \alpha\}_{j=1}^{N_E}$
  Build training set $\mathcal{D}^k = \mathcal{D}^k_E \cup \mathcal{D}^k_{RL}$
  $\widehat{\pi}_{k+1} = \arg\min_{\pi \in \Pi} \widehat{\mathcal{L}}_{\pi_k}(\hat{\rho}, \hat{\mu}, \pi)$ (Classification)
end for

tion problem, the integration of the expert and RL data sums up to merging the two datasets. Thanks to its simple form, the resulting algorithm (DPID) (Sect. 3) does not increase the complexity of DPI and its performance across iterations can be theoretically analyzed (Sect. 4). Experimental results in two simulated domains show the potential of DPID compared to existing methods (Sect. 5).

2 Preliminaries

We consider a discounted MDP defined by the tuple $(\mathcal{X}, \mathcal{A}, p, r, \gamma)$, where the state space $\mathcal{X}$ is bounded in $\mathbb{R}^d$, the action space $\mathcal{A}$ is finite[1], the kernel $p : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ defines the transition probabilities, the reward $r : \mathcal{X} \times \mathcal{A} \to [-R_{\max}; R_{\max}]$ is bounded, and $\gamma \in [0, 1)$ is a discount factor. A deterministic policy π is a mapping $\mathcal{X} \to \mathcal{A}$ and Π is a set of policies. We denote by $V^\pi : \mathcal{X} \to \mathbb{R}$ and $Q^\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ the value and action value functions of policy π. V^π is the unique fixed-point of the Bellman operator T^π defined as $(T^\pi V)(x) = r(x, \pi(x)) + \gamma \int_{\mathcal{X}} p(dy|x, \pi(x)) V(y)$, while $Q^\pi(x, a) = r(x, a) + \gamma \int_{\mathcal{X}} p(dy|x, a) V^\pi(y)$. The optimal policy $\pi^\star$ maximizes V^π over all states and we denote by $V^\star$ and $Q^\star$ its corresponding functions. $V^\star$ is also the unique fixed-point of the optimal Bellman operator T defined as $(TV)(x) = \max_a [r(x, a) + \gamma \int_{\mathcal{X}} p(dy|x, a) V(y)]$, while $Q^\star(x, a) = r(x, a) + \gamma \int_{\mathcal{X}} p(dy|x, a) V^\star(y)$. We say that π is greedy with respect to an action value function Q, if in any state x, $\pi(x) \in \arg\max_{a \in \mathcal{A}} Q(x, a)$. Finally, we introduce the greedy policy operator $\mathcal{G}$ mapping policies to greedy policies as $(\mathcal{G}\pi)(x) = \arg\max_{a \in \mathcal{A}} Q^\pi(x, a)$.

3 Direct Policy Iteration with Demonstrations

Building on *direct policy iteration* (DPI) [Lazaric *et al.*, 2010], we introduce DPI *with demonstration* (DPID), where demonstrations from an expert policy ξ are available. In the original DPI, at each iteration k, a rollout set $\{x_i\}_{i=1}^{N_{RL}}$ is sampled from a given sampling distribution ρ over $\mathcal{X}$ and the value of the current policy $\widehat{\pi}_k$ is estimated on those states using M rollouts truncated to H steps. The resulting estimates $\widehat{Q}^{\widehat{\pi}_k}(x_i, a)$ are then used to build a (cost sensitive) training set $\mathcal{D}^k_{RL} = \{x_i, a^*_i, w_i\}_{i=1}^{N_{RL}}$, where a^*_i is the best estimated action (i.e., $a^*_i = \arg\max_a \widehat{Q}^{\widehat{\pi}_k}(x_i, a)$) and $w_i = \widehat{Q}^{\widehat{\pi}_k}(x_i, a^*_i) - \widehat{Q}^{\widehat{\pi}_k}(x_i, a^+)$ is the cost of misclassifying the sample with a^+ the other action in $\mathcal{A}$. Since $\mathcal{A}$ is finite, a^*_i can be interpreted as the *label* of x_i and thus DPI reduces to solving a (binary) cost-sensitive classification problem using a given policy space Π (e.g., a linear classifier) and it returns a policy $\widehat{\pi}_{k+1}$ approximating the greedy policy $\mathcal{G}(\widehat{\pi}_k)$. In DPID we further assume access to a training set $\mathcal{D}^k_E = \{z_j, \xi(z_j)\}_{j=1}^{N_E}$, where states z_j are drawn from a distribution μ over $\mathcal{X}$ and ξ is an expert policy (not necessarily optimal). The basic idea of DPID is then to define a suitable integration of $\mathcal{D}^k_{RL}$ and $\mathcal{D}^k_E$ and solve a classification problem on the joint training set. We introduce the following elements.

Definition 1. *At each iteration k, we define the state and average empirical RL losses of a policy π as*

$$\hat{\ell}^{RL}_{\widehat{\pi}_k}(x; \pi) = \max_{a \in \mathcal{A}} \widehat{Q}^{\widehat{\pi}_k}(x, a) - \widehat{Q}^{\widehat{\pi}_k}(x, \pi(x)),$$

$$\widehat{\mathcal{L}}^{RL}_{\widehat{\pi}_k}(\widehat{\rho}, \pi) = \frac{1}{N_{RL}} \sum_{i=1}^{N_{RL}} [\max_a \widehat{Q}^{\widehat{\pi}_k}(x_i, a) - \widehat{Q}^{\widehat{\pi}_k}(x_i, \pi(x_i))],$$

where $\widehat{Q}^{\widehat{\pi}_k}$ is the rollout estimate of $Q^{\widehat{\pi}_k}$ (see Alg. 1) and $\widehat{\rho}$ is the empirical counterpart of ρ (i.e., average of Dirac distributions at observed states). We also define the state and average empirical expert losses of a policy π as

$$\ell^E(x; \pi) = \mathbb{I}\{\xi(x) \neq \pi(x)\},$$

$$\mathcal{L}^E(\widehat{\mu}, \pi) = \frac{1}{N_E} \sum_{j=1}^{N_E} \ell^E(z_j; \pi),$$

where $\widehat{\mu}$ is the empirical counterpart of μ. Finally, the average empirical joint loss is

$$\widehat{\mathcal{L}}_{\widehat{\pi}_k}(\widehat{\rho}, \widehat{\mu}, \pi) = \widehat{\mathcal{L}}^{RL}_{\widehat{\pi}_k}(\widehat{\rho}, \pi) + \alpha \mathcal{L}^E(\widehat{\mu}, \pi),$$

where $\alpha \geq 0$ is a parameter of the algorithm.

While the RL loss $\hat{\ell}^{RL}_{\widehat{\pi}_k}(x; \pi)$ depends on the policy at iteration k, the expert loss does not change through iterations since it only measures the "similarity" w.r.t. ξ using a 0/1 loss. In the following we will also use $\mathcal{L}^E_\alpha(\widehat{\mu}, \pi) = \alpha \mathcal{L}^E(\widehat{\mu}, \pi)$.

As reported in Alg. 1, DPID turns the expert training set to a cost-sensitive set by adding a constant penalty α, which allows to move from DPI ($\alpha = 0$) to a full imitation learning algorithm ($\alpha = \infty$). Once the policy $\widehat{\pi}_{k+1}$ is computed by minimizing the joint loss $\widehat{\mathcal{L}}_{\widehat{\pi}_k}$, the whole process is repeated over iterations. With DPID we propose a seamless integration of RL and expert data in a common training set, which does not affect the complexity of the original DPI algorithm. This is in contrast with APID and RLED, where the integration of expert data in the form of explicit constraints leads to optimization problems which are more complex and potentially computationally challenging (e.g., RLED needs solving a regularized non-convex optimization problem and APID defines a non-classical constrained convex optimization).

[1] In order to simplify the theoretical analysis we will consider the case $|\mathcal{A}| = 2$ throughout the rest of the paper.

4 Finite Sample Analysis

In this section we provide a complete analysis of the performance of DPID. We first define the *expected* counterparts of the losses introduced in Def. 1.

Definition 2. *At each iteration k, we define the state and average RL losses of any policy π as*

$$\ell^{RL}_{\widehat{\pi}_k}(x;\pi) = \max_{a\in\mathcal{A}} Q^{\widehat{\pi}_k}(x,a) - Q^{\widehat{\pi}_k}(x,\pi(x)),$$

$$\mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\pi) = \int_{\mathcal{X}} \Big(\max_{a\in\mathcal{A}} Q^{\widehat{\pi}_k}(x,a) - Q^{\widehat{\pi}_k}(x,\pi(x)) \Big) \rho(dx).$$

Similarly, the expert loss of any policy π is

$$\mathcal{L}^{E}(\mu,\pi) = \int_{\mathcal{X}} \ell^{E}(x;\pi)\mu(dx).$$

Finally, the joint loss is defined as

$$\mathcal{L}(\rho,\mu,\pi) = \mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\pi) + \alpha\mathcal{L}^{E}(\mu,\pi)$$

While the difference of RL loss $\mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\pi)$ w.r.t. its empirical counterpart $\widehat{\mathcal{L}}^{RL}_{\widehat{\pi}_k}(\widehat{\rho},\pi)$ is in both the sampling distribution and use of estimated Q-values, $\mathcal{L}^E$ only differs in using samples from μ. We also introduce a series of policies of interest:

$$\pi^{RL}_{k+1} = \arg\min_{\pi\in\Pi} \ \mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\pi),$$
$$\pi^{E} = \arg\min_{\pi\in\Pi} \ \mathcal{L}^{E}(\mu,\pi),$$
$$\pi_{k+1} = \arg\min_{\pi\in\Pi} \ \mathcal{L}_{\widehat{\pi}_k}(\rho,\mu,\pi).$$

These policies are the minimizers within the policy space Π of the losses defined above and thus they are the best policies in approximating the greedy policy $\mathcal{G}(\widehat{\pi}_k)$, the expert policy ξ, and the mixture obtained by averaging the two losses respectively. In the following, we assume that Π has a finite VC-dimension $h = VC(\Pi)$ and we prove a bound on the loss of the policy returned by DPID at each iteration.

Theorem 1. *Let $\widehat{\pi}_{k+1}$ be the policy returned by* DPID *and*

$$\epsilon_{RL} = 20Q_{max}\sqrt{\frac{2}{N_{RL}}\left(h\ \log\big(\frac{eN_{RL}}{h}\big) + \log\big(8/\delta\big)\right)},$$

$$\epsilon_{E} = \sqrt{\frac{8}{N_{E}}\left(h\log\big(\frac{eN_{E}}{h}\big) + \log\big(\frac{4}{\delta}\big)\right)}.$$

Then with probability $(1-6\delta)$

$$\mathcal{L}_{\widehat{\pi}_k}(\rho,\mu,\widehat{\pi}_{k+1}) \le \mathcal{L}_{\widehat{\pi}_k}(\rho,\mu,\pi_{k+1}) + 2\epsilon_{all}.$$

with $\epsilon_{all} = \epsilon_{RL} + \alpha\epsilon_E$. Furthermore,

$$\mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\widehat{\pi}_{k+1}) \le \mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\pi^{RL}_{k+1}) + 2\epsilon_{all} + \mathcal{L}^{E}_{\alpha}(\mu,\pi^{RL}_{k+1}),$$
$$\mathcal{L}^{E}_{\alpha}(\rho,\widehat{\pi}_{k+1}) \le \mathcal{L}^{E}_{\alpha}(\mu,\pi^{E}) + 2\epsilon_{all} + \mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\pi^{E}).$$

Remark (the bound). The first statement refers to the quality of $\widehat{\pi}_{k+1}$ in minimizing the joint loss and, as usual, two sources of error appear. The first is the approximation error related to the best policy in Π minimizing the loss $\mathcal{L}_{\widehat{\pi}_k}$, while the second term is the estimation error due to the limited number of samples. To reduce the estimation error, both RL and expert sample sizes should increase. The second and third statements bound the RL and expert losses of $\widehat{\pi}_{k+1}$ and compare it to the best possible policy in Π minimizing $\mathcal{L}^{RL}$ and $\mathcal{L}^{E}$ respectively. In both cases, besides the approximation and estimation errors, $\widehat{\pi}_{k+1}$ suffers from an error depending on the errors of π^{RL}_{k+1} and π^{E}_{k+1} measured w.r.t. the expert and RL losses respectively. This means that the RL loss of $\widehat{\pi}_{k+1}$ will approach the smallest loss possible within Π whenever π^{RL}_{k+1} is indeed similar to the expert policy ξ, i.e., when $\mathcal{L}^{E}_{\alpha}(\mu,\pi^{RL}_{k+1})$ is small (similar for the third statement). As expected, the behavior of DPID strictly depends also on the parameter α. First, the relevance of different estimation errors in ϵ_{all} depends on α, so that for, e.g., small α, RL errors are more relevant. Similarly, whenever α is small, $\widehat{\pi}_{k+1}$ tends to have a performance very similar to the best approximation of the greedy policy (i.e., π^{RL}_{k+1}), while when α is large $\widehat{\pi}_{k+1}$ will approach the performance of π^E.

Remark (comparison with APID). While the per-iteration error reported in [Kim *et al.*, 2013] is measured w.r.t. the policy Bellman error, here we consider different losses. While the estimation errors are similar in their dependency on N_{RL} and N_E and α, the rest of the bounds cannot be easily compared. In fact, in [Kim *et al.*, 2013] the impact of terms such as α, the similarity of ξ and $\widehat{\pi}_k$, and the approximation space Π on the performance of APID is not clearly understandable, while in DPID, thanks to its simple integration of RL and expert data, their impact is relatively straightforward.

Before deriving a full propagation analysis for DPID, we report an intermediate lemma showing how the performance loss of $\widehat{\pi}_{k+1}$ (i.e., the difference w.r.t. the optimal policy) can be upper-bounded simultaneously by the quality of the expert policy and the *standard* propagation of error.

Lemma 1. *Let V^E be the expert value function and P^π the transition kernel induced by any policy π, then at the end of iteration $k+1$, we have the state-wise bounds*

$$V^\star - V^{\widehat{\pi}_{k+1}} \le \gamma P^{\pi^\star}(V^\star - V^{\widehat{\pi}_k}) + (I - \gamma P^{\widehat{\pi}_{k+1}})^{-1}\ell^{RL}_{\widehat{\pi}_k}(\widehat{\pi}_{k+1}),$$

$$V^\star - V^{\widehat{\pi}_{k+1}} \le V^\star - V^E + \frac{2R_{max}}{1-\gamma}(I - \gamma P^{\xi})^{-1}\ell^{E}(\widehat{\pi}_{k+1}).$$

This lemma shows that the performance loss of DPID is related to two terms. The first (standard) term links the performance of $\widehat{\pi}_{k+1}$ with the previous iteration and accounts for the RL loss, measuring how well $\widehat{\pi}_{k+1}$ approximates the greedy policy $\mathcal{G}(\widehat{\pi}_k)$. The second term reveals that the performance is also related to the performance of the expert and the actual difference between $\widehat{\pi}_{k+1}$ and ξ. Thus, the actual performance loss of $\widehat{\pi}_{k+1}$ will be the minimum between these bounds, suggesting that at any iteration k DPID will benefit from either the quality of the expert or the policy improvement depending on which of the two performs the best. This means that whenever the expert is nearly optimal, $\widehat{\pi}_{k+1}$ is likely to perform as well as ξ plus the expert loss. On the other hand, if the expert is suboptimal, the standard approximate policy iteration guarantee holds.

To bound the performance loss bound over iterations, we introduce standard assumptions on the concentrability coeffi-

cients of the MDP and distributions ρ and μ, w.r.t. to the target distribution η (see [Lazaric *et al.*, 2010] for a discussion).

Assumption 1. *For any policy π and non-negative integers s and t, there exists a constant $C_{\eta,\rho}(s,t)$ such that $\eta(P^\star)^s(P^\pi)^t \leq C_{\eta,\rho}(s,t)\rho$ and $C_{\eta,\rho} = (1-\gamma)^2 \sum_{s=0}^{\infty}\sum_{t=0}^{\infty}\gamma^{s+t}C_{\eta,\rho}(s,t) < \infty$. For the expert policy ξ, there exists $C^E_{\eta,\mu}(s,t)$ such that $\eta(P^\star)^s(P^\xi)^t \leq C^E_{\eta,\mu}(s,t)\mu$ and $C^E_{\eta,\mu} = \sum_{t=0}^{\infty}\gamma^{s+t}C^E_{\eta,\mu}(s,t) < \infty$.*

Theorem 2. *Let $\Delta_k = V^\star - V^{\widehat{\pi}_k}$ and $\Delta_E = V^\star - V^E$ be the performance loss of the algorithm and the expert respectively. Under Assumption 1, if* DPID *is run over K iterations then*

$$\|\Delta_K\|_{1,\eta} \leq \min\left\{A_K, B_K, C_K\right\},$$

with

$$A_K = \|\Delta_E\|_{1,\eta} + \frac{2R_{max}}{(1-\gamma)^2}C_{\eta,\mu}\mathcal{L}^E(\mu,\widehat{\pi}_K),$$

$$B_K = \min_j\left\{\gamma^{K-j}\|\Delta_E\|_{1,\eta} + \frac{2\gamma^{K-j}}{(1-\gamma)^2}C^E_{\eta,\mu}\mathcal{L}^E(\mu,\widehat{\pi}_j) + \frac{C_{\eta,\rho}}{(1-\gamma)^2}\max_{l=j,\dots,K-1}\mathcal{L}^{RL}_{\widehat{\pi}_l}(\rho,\widehat{\pi}_{l+1})\right\},$$

$$C_K = \frac{\gamma^K}{1-\gamma} + \frac{C_{\eta,\rho}}{(1-\gamma)^2}\max_{k=1,\dots,K-1}\mathcal{L}^{RL}_{\widehat{\pi}_l}(\rho,\widehat{\pi}_{k+1}).$$

Remark (propagation bound). The performance loss is the minimum over three terms. The term C_K is the same as in the propagation error of DPI and it depends on the largest policy improvement error over iterations. The term A_K depends on the quality of the expert and the expert loss of the last policy $\widehat{\pi}_K$. The most interesting term is B_K, which gives an important insight on how DPID can actually benefit from the repeated balancing of expert and RL data. In fact, whenever there is an iteration j in which the combined RL and expert losses are small, this will lead to an overall small performance loss. Finally, combining Thm. 1 and 2, we can obtain the global *finite-sample* bound. The resulting bound (omitted for the sake of space) shows the critical impact of α and the quality of ξ in the final performance. In fact, whenever the ξ is optimal (i.e. $\Delta_E = 0$) and α is large, the second statement in Thm. 1 guarantees that DPID performs almost as well as the policy π^E which approximates ξ the best within the policy space Π (term A_K is the bound). On the other hand, if ξ is highly suboptimal and α is small, then DPID would approach the performance of DPI (term C_K in the bound). In all other intermediate cases, DPID could still effectively take advantage of both the DPI process and the imitation of ξ to achieve a nearly optimal performance (term B_K in the bound).

5 Experiments

We compare the performance of DPID[2] to DPI, which uses only RL data, and to DAGGER [Ross *et al.*, 2010] when the expert can be queried at arbitrary states or pure imitation learning when batch expert data are available. We also compare to

[2]The implementation of DPID is available at https://www.dropbox.com/s/jj4g9ndonol4aoy/dpid.zip?dl=0

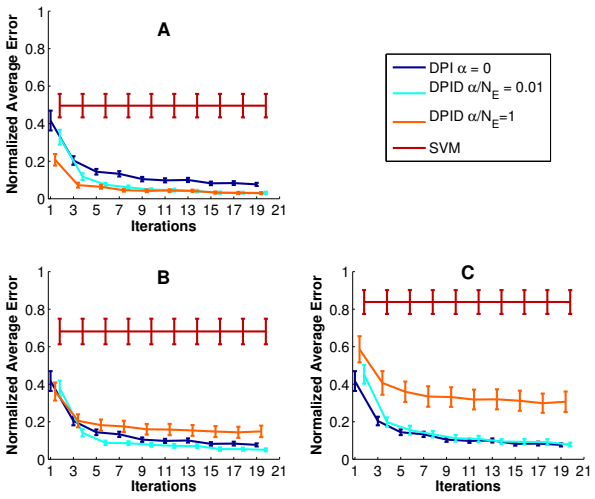

Figure 1: *Panel A*: Fixed $N_E = 15$ optimal expert demonstrations and increasing N_{RL} by 50 at each iteration, starting with $N_{RL} = 50 - N_E$ ($N_E = 0$ for DPI). *Panel B and C*: Same setting but expert with non-optimal actions with probability 0.25 (B) or 0.5 (C).

APID (implemented using CVX [Grant and Boyd, 2014]) and RLED (implemented using decision trees as weak learners).

Garnet domain. The Garnet framework [Archibald *et al.*, 1995; Bhatnagar *et al.*, 2009; Piot *et al.*, 2014] allows to generate random finite MDPs characterized by 3 parameters: number of states N_s, number of actions N_a, and maximum branching factor N_b. We construct the transition function $p(x,a,x')$ to mimic dynamical systems where an action moves the state to states that are close by. Given a state x, next states are generated by sampling N_b times from a Gaussian distribution centered at x. The probabilities of going to each of the next states are chosen uniformly at random from $[0,1]$ and then normalized to 1. For every action a, we construct the reward $r(x,a)$ by sampling at random 20% of the states and randomly assigning them rewards of 1 or -1. Everywhere else the reward is 0. For such MDPs, we compute $\pi^\star$ and $V^\star$ using exact policy iteration and we measure the performance of the policy π_{alg} returned by any of the algorithms as the *normalized average error* $Err_{alg} = \frac{\mathbb{E}[V^* - V^{\pi_{alg}}]}{\mathbb{E}[V^*]}$. In the following, we use $N_s = 15$, $N_a = 3$, $N_b(s) \in [3,6]$ and we estimate Err_{alg} over 100 independent runs, while error bars are computed as 95% Gaussian confidence intervals.

We start with a set of experiments where the exact state representation is used (i.e., no approximation error). This means that all algorithms have the potential to reach a zero error as the number of samples increases thus avoiding confounding effects due to the specific choice of feature design. In Fig.1*A* we compare DPID with DPI and pure imitation learning (implemented with SVM [Chang and Lin, 2011]) when the expert is optimal and number of demonstrations is small ($N_E = 15$) and fixed, while the RL samples are increased at every iteration by 50. While DPI and DPID both perform better than imitation learning, DPID achieves the best performance thanks to the positive bias provided by the expert demonstrations. As expected, the greater the weight of ex-

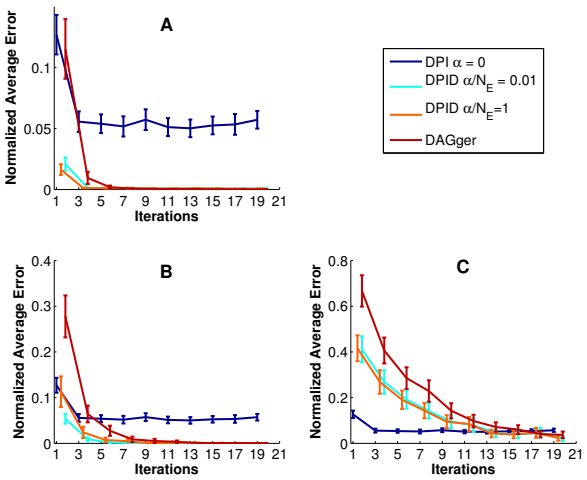

Figure 2: $N = 1500$. *Panel A*: $N_{RL} = 1500 - N_E$ for DPID with N_E optimal expert demonstrations increases by 50 at each iteration. $N_{RL} = 1500$ for DPI. *Panel B and C*: Same setting but expert with non-optimal action with probability 0.25 (B) or 0.5 (C).

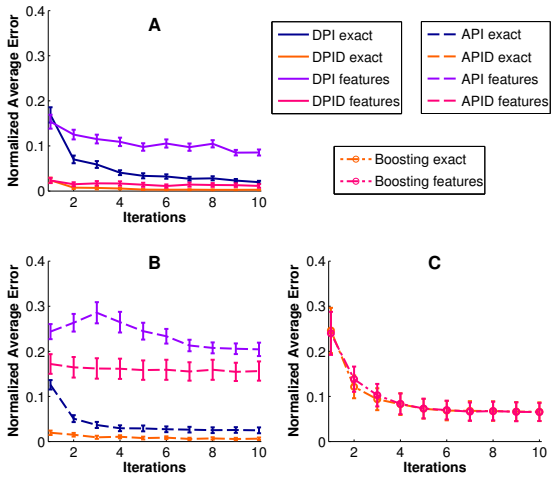

Figure 3: Fixed $N_E = 50$ optimal expert samples and N_{RL} increasing over iterations by 500, starting with $N_{RL} = 500$-N_E ($N_E = 0$ for DPI and API). *Panel A*: DPI and DPID with exact and approximate representation. *Panel B-C:* Same for API, APID, and RLED.

pert demonstrations the stronger the bias in initial iterations, although they later converge to the same performance. In Fig.1B and C, we replace the optimal expert with a suboptimal one by sampling random non-optimal actions 25% and 50% of the time respectively. Since α rules the trade-off between RL and expert losses, we can see that depending on α, DPID performs either better (B, α small), similarly (C, α small), or worse (B and C, α large) than DPI. In Fig.2A we fix the total number of samples $N = N_{RL} + N_E = 1500$, but change the proportion of expert to RL samples, starting from $N_E = 50$ up to 1000. As the number of expert samples increases, DPID and DAGGER converge to the optimal policy, however DPI reaches a plateau due to the variance caused by the limited number of samples. In Fig. 2B and C we replace the optimal expert with a suboptimal one. All algorithms have errors that decrease with increasing expert samples. However, DAGGER performs the worst and DPID with a small α converges the best. In B the expert is good enough to guide towards the optimal policy, while in C it induces a bias and slows down the convergence.

We now introduce an approximate representation such that for every state, we construct a binary feature vector of length $d = 6 < N_s$. The number of ones in the representation is set to $l = 3$ and their locations are chosen randomly as in [Bhatnagar *et al.*, 2009]. Fig. 3A compares the performance of DPI and DPID with or without features in the same setting as Fig. 1A. We notice that DPI is significantly affected by the bias introduced by the features, but DPID is barely so and converges to the optimal policy. In B we compare LSPI (which corresponds to APID with $\alpha = 0$) and APID. Here both algorithms are affected by the bias, but APID is still better than LSPI in both cases. In C we show the performance of RLED, which does not use our representation since it constructs features automatically even in the exact representation case. We note the different effect that feature design had on the different algorithms, with DPID being the most robust.

Vehicle Brake Control domain. In this domain, the state space is continuous and consists of a 4-dimensional vector representing the current velocity, the acceleration pedal value, the brake pedal value, and the target velocity. The goal of the agent is to reach the target velocity and maintain it. There are 5 actions available: *acceleration up, acceleration down, brake up, brake down, do nothing*. The reward equals -10 times the absolute difference between the current and the target velocity. As in [Kim *et al.*, 2013] the expert is implemented using the dynamics of the pedal pressure and the velocity output with the addition of random noise. We compare the performances of DPID, APID, and RLED. As in [Kim *et al.*, 2013] we use Radial Basis Functions to represent the state in APID. Fig. 4A and D are the DPID experiments and show the results of using an optimal and suboptimal expert respectively. This domain turns out to be easy for our classification-based algorithm which converges quickly to a very good policy. In D, a larger α hurts the performance possibly because the expert biases the sampling process and renders RL useless. B and E show the results of APID, which seems to be less suited to this problem and converge very slowly. Expert samples help in both cases, with larger α performing better. Finally, C and F are the results of RLED. The performance is similar to that of APID which is unsurprising because they are both estimating the same quantity. Surprisingly, a larger α leads to worse performance in both cases. Although the comparison among these methods is partially biased by the difference in the underlying learning schemes (classification-based vs projected Bellman residual minimization vs optimal Bellman residual minimization), these results show that DPID is very effective in integrating RL and expert data and it is relatively robust to expert sub-optimality and poor choice of α.

6 Conclusion

In this paper, we introduced DPID that naturally integrates RL and expert data. We provided a full analysis of the propa-

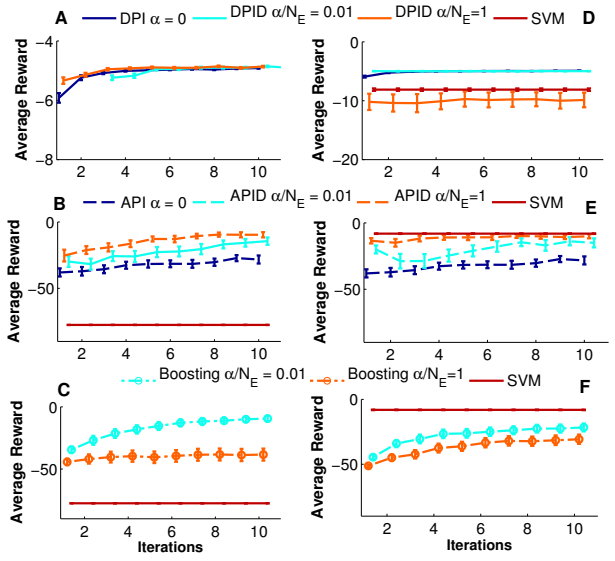

Figure 4: *Panel A*: DPI versus DPID when fixing $N_E = 15$ optimal expert demonstrations and increasing N_{RL} by 500 at every iteration. *Panel D*: DPI versus DPID when fixing $N_E = 100$ suboptimal expert demonstrations and increasing the RL data by 500 at iteration (0.5 prob. of wrong action). *Panels B/E and C/F*: Same as A and D for API/APID and RLED respectively.

gation error which resulted in insightful bounds. In particular, the impact of α suggests that a natural improvement on DPID would be to adapt α over iterations depending on the quality of the current policy compared to the expert policy. A principled solution may take inspiration from conservative policy iteration [Kakade and Langford, 2002], where monotonic policy improvement is guaranteed at each iteration. Finally, we showed empirically that even a few or suboptimal expert demonstrations can lead to significant improvements and that DPID compares favorably to APID and RLED.

Acknowledgments

This work was supported by the French Ministry of Higher Education and Research and the French National Research Agency (ANR) under project ExTra-Learn n.ANR-14-CE24-0010-01.

A Proofs

We first introduce two technical lemma whose proofs follow exactly the same steps as in Lemma 1 of [Lazaric *et al.*, 2010].

Lemma 2. *Let $\{x_i\}_{i=1}^{N_{RL}}$ be a rollout set sampled from ρ. If in each state we simulate M truncated rollouts, then*

$$\mathbb{P}\left[\sup_{\pi\in\Pi}\left|\frac{1}{N_{RL}}\sum_{i=1}^{N_{RL}}\frac{1}{M}\sum_{j=1}^{M}R_j^{\widehat{\pi}_k}(x_i,\pi(x_i)) - \frac{1}{N_{RL}}\sum_{i=1}^{N_{RL}}Q^{\widehat{\pi}_k}(x_i,\pi(x_i))\right| > \epsilon_{2_{RL}}\right] \le \delta,$$

with $\epsilon_{2_{RL}} = \frac{8(1-\gamma^H)}{\sqrt{MN_{RL}}}Q_{max}\sqrt{2(h\log(\frac{eMN_{RL}}{h})+\log(\frac{8}{\delta}))}$ *and*

$$\mathbb{P}\left[\sup_{\pi\in\Pi}|\mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\pi) - \mathcal{L}^{RL}_{\widehat{\pi}_k}(\hat{\rho},\pi)| > \epsilon_{1_{RL}}\right] \le \delta,$$

with $\epsilon_{1_{RL}} = 16Q_{max}\sqrt{\frac{2}{N_{RL}}(h\ \log(\frac{eN_{RL}}{h}) + \log(\frac{8}{\delta}))}$.

Lemma 3. *Let $\{z_j\}_{j=1}^{N_E}$ be a set sampled from μ, then*

$$\mathbb{P}\left[\sup_{\pi\in\Pi}|\mathcal{L}^{E}(\hat{\mu},\pi) - \mathcal{L}^{E}(\mu,\pi)| > \epsilon_E\right] \le \delta, \tag{1}$$

with $\epsilon_E = \sqrt{\frac{8}{N_E}(h\log(\frac{eN_E}{h}) + \log(\frac{4}{\delta}))}$.

Proof of Theorem 1.

$$\begin{aligned}
&\mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\hat{\pi}_{k+1}) + \alpha\mathcal{L}^{E}_{\widehat{\pi}_k}(\mu,\hat{\pi}_{k+1})\\
&\le \hat{\mathcal{L}}^{RL}_{\widehat{\pi}_k}(\hat{\rho},\hat{\pi}_{k+1}) + \alpha\mathcal{L}^{E}_{\widehat{\pi}_k}(\hat{\mu},\hat{\pi}_{k+1}) + \epsilon_{1_{RL}} + \epsilon_{1_E} + \epsilon_{2_{RL}}\\
&\le \hat{\mathcal{L}}^{RL}_{\widehat{\pi}_k}(\hat{\rho},\pi) + \alpha\mathcal{L}^{E}_{\widehat{\pi}_k}(\hat{\mu},\pi) + \epsilon_{1_{RL}} + \epsilon_{1_E} + \epsilon_{2_{RL}}\\
&\le \mathcal{L}^{RL}_{\widehat{\pi}_k}(\rho,\pi) + \alpha\mathcal{L}^{E}_{\widehat{\pi}_k}(\mu,\pi) + 2\epsilon_{all}
\end{aligned} \tag{2}$$

where $\epsilon_{all} = \epsilon_{1_{RL}} + \epsilon_{1_E} + \epsilon_{2_{RL}}$. The last inequality holds for any policy π, in particular π^{RL}_{k+1} or π^{E}_{k+1} which completes the proof. □

Proof of Lemma 1. The second statement is proved in Section 4.2 of [Lazaric *et al.*, 2010], while the first statement is obtained from

$$\begin{aligned}
V^{\xi} - V^{\pi_{k+1}} &= T^{\xi}V^{\xi} - T^{\xi}V^{\pi_{k+1}} + T^{\xi}V^{\pi_{k+1}} - V^{\pi_{k+1}}\\
&= \gamma P^{\xi}(V^{\xi} - V^{\pi_{k+1}}) + T^{\xi}V^{\pi_{k+1}} - V^{\pi_{k+1}}\\
&= (I-\gamma P^{\xi})^{-1}(T^{\xi}V^{\pi_{k+1}} - V^{\pi_{k+1}})\\
&= (I-\gamma P^{\xi})^{-1}(Q^{\pi_{k+1}}(\xi) - Q^{\pi_{k+1}}(\pi_{k+1}))\\
&\le \frac{2R_{\max}}{1-\gamma}(I-\gamma P^{\xi})^{-1}\mathbb{I}(\xi \ne \pi_{k+1})\\
&= \frac{2R_{\max}}{1-\gamma}(I-\gamma P^{\xi})^{-1}\ell^{E}(\pi_{k+1}).
\end{aligned}$$

□

Proof of Theorem 2. We rewrite the right-hand side of the first statement of Lem. 1 as

$$A_k = (V^{\star} - V^{E}) + \frac{2R_{\max}}{1-\gamma}(I-\gamma P^{\xi})^{-1}\ell^{E}(\widehat{\pi}_k),$$

and we introduce

$$E_k = (I-\gamma P^{\pi_{k+1}})^{-1} \quad \text{and} \quad \Delta_{k+1} = V^{\star} - V^{\widehat{\pi}_{k+1}}.$$

Then Lem. 1 can be written in a more compact way as

$$\Delta_{k+1} \le \min\left\{A_{k+1}; \gamma P^{\star}\Delta_k + E_k\ell^{RL}_{\widehat{\pi}_k}(\widehat{\pi}_{k+1})\right\}.$$

Now we can apply Lem. 1 recursively and obtain

$$\begin{aligned}
\Delta_{k+1} &\le \min\Big\{A_{k+1};\\
&\gamma P^{*}\min\left\{A_k; \gamma P^{\star}\Delta_{k-1} + E_{k-1}\ell^{RL}_{\widehat{\pi}_{k-1}}(\widehat{\pi}_k)\right\} + E_k\ell^{RL}_{\widehat{\pi}_k}(\widehat{\pi}_{k+1})\Big\}\\
&= \min\Big\{A_{k+1};\quad \gamma P^{*}A_k + E_k\ell^{RL}_{\widehat{\pi}_k}(\widehat{\pi}_{k+1});\\
&\qquad (\gamma P^{*})^2\Delta_{k-1} + \gamma P^{*}E_{k-1}\ell^{RL}_{\widehat{\pi}_{k-1}}(\widehat{\pi}_k) + E_k\ell^{RL}_{\widehat{\pi}_k}(\widehat{\pi}_{k+1})\quad\Big\}.
\end{aligned}$$

At each application of Lem. 1, we generate additional elements similar to the second term in the previous equation, and thus

$$\begin{aligned}
\Delta_{k+1} \le \min\Big\{&A_{k+1};\\
&\min_{j=1,\dots,k}\left\{(\gamma P^{*})^{k+1-j}A_j + \sum_{l=j}^{k}(\gamma P^{*})^{k+1-l}E_l\ell^{RL}_{\pi_l}(\pi_{l+1})\right\};\\
&(\gamma P^{*})^{k+1}\Delta_0 + \sum_{l=0}^{k}(\gamma P^{*})^{k-l}E_l\ell^{RL}_{\pi^l}(\pi_{l+1})\Big\}.
\end{aligned}$$

As a direct application of the previous inequality, we bound the absolute value of the performance loss of DPID after K iterations and then take the η-norm, thus concluding the proof. □

References

[Abbeel and Ng, 2004] Peter Abbeel and Andrew Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[Archibald *et al.*, 1995] TW Archibald, KIM McKinnon, and LC Thomas. On the generation of markov decision processes. *Journal of the Operational Research Society*, pages 354–361, 1995.

[Bagnell and Ross, 2010] J Andrew Bagnell and Stephane Ross. Efficient Reductions for Imitation Learning. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 661–668, 2010.

[Bertsekas and Tsitsiklis, 1996] Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition, 1996.

[Bhatnagar *et al.*, 2009] Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Doshi-Velez *et al.*, 2010] Finale Doshi-Velez, David Wingate, Nicholas Roy, and Joshua B Tenenbaum. Nonparametric bayesian policy priors for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 532–540, 2010.

[Farahmand *et al.*, 2009] Amir M Farahmand, Mohammad Ghavamzadeh, Shie Mannor, and Csaba Szepesvári. Regularized policy iteration. In *Advances in Neural Information Processing Systems*, pages 441–448, 2009.

[Grant and Boyd, 2014] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[Kakade and Langford, 2002] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 267–274, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[Kim *et al.*, 2013] Beomjoon Kim, Amir massoud Farahmand, Joelle Pineau, and Doina Precup. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems*, pages 2859–2867, 2013.

[Lagoudakis and Parr, 2003] Michail G Lagoudakis and Ronald Parr. Reinforcement learning as classification: Leveraging modern classifiers. In *ICML*, volume 3, pages 424–431, 2003.

[Lazaric *et al.*, 2010] Alessandro Lazaric, Mohammad Ghavamzadeh, Rémi Munos, et al. Analysis of a classification-based policy iteration algorithm. In *ICML-27th International Conference on Machine Learning*, pages 607–614, 2010.

[Munos, 2003] R. Munos. Error bounds for approximate policy iteration. In *International Conference on Machine Learning*, pages 560–567, 2003.

[Ng and Russell, 2000] A. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 663–670, 2000.

[Piot *et al.*, 2014] Bilal Piot, Matthieu Geist, and Olivier Pietquin. Boosted Bellman Residual Minimization Handling Expert Demonstrations. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*. Springer, 2014.

[Ross and Bagnell, 2014] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv:1406.5979*, 2014.

[Ross *et al.*, 2010] Stéphane Ross, Geoffrey J Gordon, and J Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *arXiv preprint arXiv:1011.0686*, 2010.